Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization

Haoran Li[1,2], **Junnan Zhu**[1,2], **Jiajun Zhang**[1,2] and **Chengqing Zong**[1,2,3]
[1] National Laboratory of Pattern Recognition, CASIA, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology
{haoran.li, junnan.zhu, jjzhang, cqzong}@nlpr.ia.ac.cn

Abstract

In this paper, we investigate the sentence summarization task that produces a summary from a source sentence. Neural sequence-to-sequence models have gained considerable success for this task, while most existing approaches only focus on improving word overlap between the generated summary and the reference, which ignore the correctness, i.e., the summary should not contain error messages with respect to the source sentence. We argue that correctness is an essential requirement for summarization systems. Considering a correct summary is semantically entailed by the source sentence, we incorporate entailment knowledge into abstractive summarization models. We propose an entailment-aware encoder under multi-task framework (i.e., summarization generation and entailment recognition) and an entailment-aware decoder by entailment Reward Augmented Maximum Likelihood (RAML) training. Experimental results demonstrate that our models significantly outperform baselines from the aspects of informativeness and correctness.

1 Introduction

Sentence summarization is a well-studied task that creates a condensed version of a long source sentence. Sequence-to-sequence (seq2seq) model that encodes a source sequence into a latent representation and outputs another sequence is the dominating framework for sentence summarization (Rush et al., 2015; Chopra et al., 2016; Takase et al., 2016; Zhou et al., 2017; Li et al., 2017b; Li et al., 2018). Despite substantial improvements on this task, most of the existing researches typically aim to improve word overlap between the generated summary and the references, which is measured by n-gram matching metrics (e.g., ROUGE (Lin, 2004)). Hence, it cannot guarantee the semantic correctness of the summary as a whole. Therefore, in some cases, the summary giving high matching scores may contain critical error messages, which makes the summary fail to capture the correct information with respect to the source sentence. Previous study shows that about 30% of the summaries generated by state-of-the-art seq2seq system are subject to this problem (Cao et al., 2017). Here is an example (the digits are replaced by "#"):

Source sentence: franch won the gold medal at women 's epee team event of the fie #### world championships by beating china ##-## .
Reference: france beats china for women 's epee team gold
State-of-the-art seq2seq model: canada wins women 's epee team event

For the example shown above, the seq2seq system produces a fluent summary which contains an obvious mistake. The true winner of the "women 's epee team event" is "france", while the summarization model wrongly generates "canada", which is probably due to similar word representations for country names. Though the word overlap between the generated summary and the reference is considerable, leading to high ROUGE scores, the summary is invalid.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1430–1441
Santa Fe, New Mexico, USA, August 20-26, 2018.

We argue that correctness is an essential requirement for summarization systems, while most existing systems ignore it. Generally, a correct summary is semantically entailed by the source sentence, thus we believe entailment[1] (Bos and Markert, 2005) knowledge is beneficial to avoid producing contradictory or unrelated information in the summary.

To incorporate entailment knowledge into abstractive summarization models, we propose in this work an entailment-aware encoder and an entailment-aware decoder. We share the encoder of the summarization generation system with the entailment recognition system, so that the encoder can grasp both the gist of the source sentence and be aware of entailment relationships. Furthermore, we propose an entailment Reward Augmented Maximum Likelihood (RAML) (Norouzi et al., 2016) training that encourages the decoder of the summarization system to produce summary entailed by the source. Experimental results demonstrate that our models significantly outperform some solid baselines on objective evaluation for informativeness and manual evaluation for correctness. Further analysis suggests that our summarization model is aware of entailment knowledge.

Our main contributions are as follows:

- We incorporate entailment knowledge into summarization models to avoid producing unrelated information with respect to the source sentence.
- We propose an entailment-aware encoder by jointly modeling summarization generation and entailment recognition.
- We introduce an entailment-aware decoder via entailment RAML training.

2 Background: Seq2seq Learning

In this section, we describe the basic seq2seq learning framework. Given a dataset of input-output pairs, $\mathcal{D} \triangleq (\mathbf{x}_i, \mathbf{y}_i^*)_{i=1}^N$, the seq2seq model maximizes the conditional probability of a target sequence $\mathbf{y}^*$: $p(\mathbf{y}^*|\mathbf{x})$. Recurrent Neural Networks (RNN) encoder (Cho et al., 2014) reads and converts a variable-length input sequence $\mathbf{x}$ into a context representation c as follows:

$$h_t = f_{\text{enc}}(\mathrm{x_t}, h_{t-1}) \tag{1}$$

$$c_t = f_c(h_1, \cdots, h_t) \tag{2}$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t, and c_t is a context vector generated from the sequence of the hidden states. f_{enc} and f_{c} are nonlinear activation functions.

The decoder generates word $\mathrm{y_t}$ given the context vector c_t and the previously generated words:

$$p(\mathrm{y}_t|\{\mathrm{y}_1, \cdots, \mathrm{y}_{t-1}\}, c_t) = f_{\text{dec}}(\mathrm{y}_{t-1}, s_t, c_t) \tag{3}$$

where s_t is the hidden state of the decoder and f_{dec} is a nonlinear activation function. The maximum likelihood (ML) framework tries to minimize negative log-likelihood of the parameters as follows:

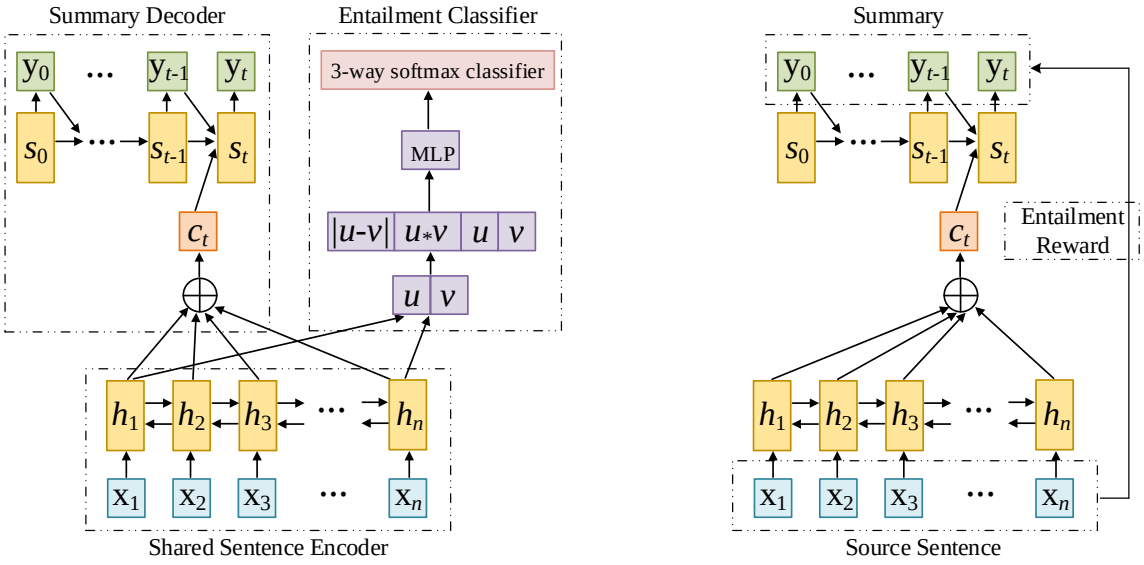
$$\mathcal{L}_{\text{ML}}(\mathcal{D}) = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} -\log p(\mathbf{y}^*|\mathbf{x}) \tag{4}$$

3 Our Proposed Model

3.1 Overview

In order to avoid generating unrelated summary with respect to the source sentence, we propose two strategies to incorporate entailment knowledge into seq2seq summarization model. We first introduce an entailment-aware encoder using multi-task learning for summarization generation and entailment recognition. Then, we introduce an entailment-aware decoder by entailment RAML training.

[1]Entailment is a kind of relationships between two sentences for natural language inference. Sentence A entailing sentence B means A can infer B. Other relationships include contradiction and neutral. A correct summary should be inferred by the source sentence. Thus, we argue that entailment is a useful criterion for the correctness of the summary.



(a) Entailment-aware Encoder

(b) Entailment-aware Decoder

Figure 1: The framework of our model. Entailment-aware encoder is learned by jointly training summarization generation (left part of (a), which is a seq2seq model) and entailment recognition (right part of (a), in which sentence pair in the entailment recognition corpus are encoded as u and v). Entailment-aware decoder is learned by entailment RAML training, in which the summary will be rewarded if it is entailed by the source sentence.

3.2 Entailment-aware Encoder

In this section, we propose a multi-task learning for abstractive summarization by sharing the encoder with the task of entailment recognition. By doing so, we can learn an entailment-aware encoder for sentence summarization task. In this way, we can improve the correctness aspect of the summarization model, while maintaining the salient information extraction aspects. Note that the training data for summarization and entailment task is from summarization and entailment corpus, respectively.

3.2.1 Shared Sentence Encoder

Given a source sentence $\mathbf{x} = (\mathrm{x}_1, \cdots, \mathrm{x}_n)$, we employ a bidirectional LSTM (BiLSTM) to build its hidden representation $(h_1, \cdots, h_n)$.

The BiLSTM encodes source sentence forwardly and backwardly to generate two sequences of the hidden states: $(\overrightarrow{h}_1, \cdots, \overrightarrow{h}_n)$ and $(\overleftarrow{h}_1, \cdots, \overleftarrow{h}_n)$, respectively, where:

$$\overrightarrow{h}_i = \mathrm{LSTM}(\mathrm{x}_i, \overrightarrow{h}_{i-1}) \tag{5}$$

$$\overleftarrow{h}_i = \mathrm{LSTM}(\mathrm{x}_i, \overleftarrow{h}_{i+1}) \tag{6}$$

The final sentence representation h_i is the concatenation of the forward and backward vectors: $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$.

3.2.2 Attention-based Summarization Decoder

At each time step t, the state of the decoder s_t is calculated as follows:

$$s_t = \mathrm{LSTM}(s_{t-1}, \mathrm{y}_{t-1}, c_t) \tag{7}$$

$$s_0 = \tanh(\mathbf{W}_h[\overrightarrow{h}_n; \overleftarrow{h}_1]) \tag{8}$$

We compute the context vector c_t as a weighted sum of the source annotations as follows:

$$c_t = \sum_{i=1}^{N} \alpha_{t,i} h_i \tag{9}$$

where each vector is weighted by the attention weight $\alpha_{t,i}$, as calculated in Equations 10 and 11:

$$e_{t,i} = v_c^T \tanh(\mathbf{W}_s s_t + \mathbf{W}_e h_i) \tag{10}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{N} \exp(e_{t,j})} \tag{11}$$

The probability for the next target word y_t is computed using hidden state s_t and the previously emitted word y_{t-1} as follows:

$$p(\mathrm{y}_t|\{\mathrm{y}_1, \cdots, \mathrm{y}_{t-1}\}) \propto \exp(\mathbf{L}_e \tanh(\mathbf{L}_s s_t + \mathbf{L}_y \mathrm{y}_{t-1})) \tag{12}$$

where $\mathbf{W}_h$, $\mathbf{W}_s$, $\mathbf{W}_e$, $\mathbf{L}_e$, $\mathbf{L}_s$ and $\mathbf{L}_y$ are model parameters. The summarization model is trained by minimizing negative log-likelihood loss as in Equation 4.

3.2.3 Matching-based Entailment Inference Model

To infer entailment relation, input sentence pairs from the entailment recognition corpus are fed into sentence encoder to obtain hidden representation $(h_1^u, \cdots, h_n^u)$ and $(h_1^v, \cdots, h_n^v)$, respectively. Then, the sentence pairs are encoded as vectors $u = [\overrightarrow{h}_n^u; \overleftarrow{h}_1^u]$ and $v = [\overrightarrow{h}_n^v; \overleftarrow{h}_1^v]$, respectively. Next, the absolute difference and the element-wise product for the tuple $[u, v]$ are concatenated with the original vectors u and v (Mou et al., 2016) as follows:

$$q = [|u - v|; u * v; u; v] \tag{13}$$

We then feed q into a 3-layer multilayer perceptron (MLP) classifier. The 3-class softmax output layer is on top of MLP. The entailment recognition model is trained by minimizing cross-entropy loss.

3.2.4 Multi-Task Learning (MTL)

In our multi-task setup, we share the encoder parameters of both the tasks, as shown in Figure 1(a). Traditional MTL considers equal contribution for all tasks. In our model, two tasks are significantly different. The summary generation task is much more complicated than entailment recognition, leading to different learning difficulties and convergence rates. Therefore, summarization generation is regarded as the main task and entailment recognition as the auxiliary task, and our goal is to optimize the main task with assistance of auxiliary task. To this end, we optimize the two loss functions alternatively during training. Let α be the number of mini-batches of training for entailment recognition after 100 mini-batches of training for summarization generation (Pasunuru et al., 2017). We adopt $\alpha = 10$ and performance with different α is discussed in Section 6.6.3.

3.3 Entailment-aware Decoder

In order to encourage the decoder of the summarization system to produce summary entailed by the source sentence, we apply an entailment-aware decoder by entailment RAML training (Norouzi et al., 2016).

3.3.1 Reward Augmented Maximum Likelihood (RAML) Training

RAML provides a computationally efficient approach to optimize task-specific reward (loss) directly. In our work, we apply RAML to incorporate entailment-based reward into our summarization model, as shown in Figure 1(b).

The RAML objective function is defined as follows:

$$\mathcal{L}_{\mathrm{RAML}} = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} \left\{ -\sum_{\mathbf{y}\in\mathcal{Y}} q\left(\mathbf{y}|\mathbf{x}, \mathbf{y}^*; \tau\right) \log p(\mathbf{y}|\mathbf{x}) \right\} \tag{14}$$

$$q\left(\mathbf{y}|\mathbf{x}, \mathbf{y}^*; \tau\right) = \frac{1}{Z(\mathbf{x}, \mathbf{y}^*, \tau)} \exp\{r(\mathbf{x}, \mathbf{y}, \mathbf{y}^*)/\tau\} \tag{15}$$

$$Z(\mathbf{x}, \mathbf{y}^*, \tau) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp\{r(\mathbf{x}, \mathbf{y}, \mathbf{y}^*)/\tau\} \tag{16}$$

where $\mathcal{Y}$ is the set of possible model outputs. $r(\mathbf{x}, \mathbf{y}, \mathbf{y}^*)$ denotes the reward function and τ is the regularization parameter.

3.3.2 Optimizing by Entailment-based Sampling

We can express the gradient of $\mathcal{L}_{\mathrm{RAML}}$ in terms of an expectation over samples from $q(\mathbf{y}|\mathbf{x}, \mathbf{y}^*; \tau)$:

$$\mathcal{L}_{\mathrm{RAML}} = E_{q(\mathbf{y}|\mathbf{x},\mathbf{y}^*;\tau)}\left[-\nabla \log p(\mathbf{y}|\mathbf{x})\right] \tag{17}$$

RAML training adds a sampling step over typical ML objective. Instead of optimizing ML on training samples, given training input $(\mathbf{x}, \mathbf{y}^*)$, RAML training first samples an output $\mathbf{y}$ proportionally to the reward. Then, RAML optimizes log-likelihood on such sample given the corresponding input. Thus, we need to sample auxiliary outputs from the exponentiated payoff distribution, $q(\mathbf{y}|\mathbf{x}, \mathbf{y}^*; \tau)$. In this work, we first use reward values defined by negative Hamming distance and then re-weight the reward based on entailment reward $s(\mathbf{x}, \mathbf{y}, \mathbf{y}^*)$. Particularly, given a sentence $\mathbf{y}^*$ of length ℓ, we count the number of sentences within an edit distance d, where $d \in \{0, \ldots, 2\ell\}$. Then, we weight the counts by $\exp\{-d/\tau\}$ and perform normalization. Finally, we apply importance sampling by the weight $\exp\{(s(\mathbf{x}, \mathbf{y}, \mathbf{y}^*) + d)/\tau\}$ and perform normalization, where the proposal distribution is Hamming distance sampling[2].

We define entailment reward $s(\mathbf{x}, \mathbf{y}, \mathbf{y}^*)$ as follows:

$$s(\mathbf{x}, \mathbf{y}, \mathbf{y}^*) = \min\{e(\mathbf{x}, \mathbf{y}), e(\mathbf{x}, \mathbf{y}^*)\} \tag{18}$$

where $e(\mathbf{x}, \mathbf{y})$ denotes entailment score for sentence pairs $(\mathbf{x}, \mathbf{y})$. Our goal is to maximize the entailment reward of the summary towards the reference, given the source sentence. Here we adopt the model of Parikh et al. (2016) trained on the MultiNLI corpus[3] (Williams et al., 2017) to obtain $e(\mathbf{x}, \mathbf{y})$.

4 Related work

Text summarization methods can be categorized into extraction-based methods (Erkan and Radev, 2004; Wan et al., 2007; Cheng and Lapata, 2016; Zhang et al., 2016; Nallapati et al., 2017; Li et al., 2017a) and abstraction-based methods. Rush et al. (2015) first apply the seq2seq model to abstractive sentence summarization. They propose an attentive CNN encoder and a neural network language model (Bengio et al., 2003) decoder. Chopra et al. (2016) use RNN as the decoder and achieve better performance. Nallapati et al. (2016) further replace the encoder with an RNN, forming a full RNN seq2seq model. Gu et al. (2016) and Zeng et al. (2016) incorporate a copying mechanism into seq2seq learning and Gulcehre et al. (2016) propose a switch gate to control whether to copy from the source or generate a word by the decoder. Copying mechanism intends to replicate segments in the source to the target, which cannot guarantee the correctness of the summary as a whole. Ma et al. (2017) focus on improving the semantic relevance between source and summary by encouraging high similarity of their representation. Zhou et al. (2017) employ a selective encoding model to control the information flow from encoder to decoder. Li et al. (2017b) apply a deep recurrent generative decoder to seq2seq framework. Cao et al. (2017) solve the problem of fake facts in a summary. They use Open Information Extraction to extract fact descriptions in the source sentence and propose the dual-attention seq2seq framework to force the generation conditioned on both source sentence and the fact descriptions. To the best of our knowledge, our work is the first to directly explore the correctness of summary without any preprocessing.

Some previous work (Mehdad et al., 2013; Gupta et al., 2014) has used textual entailment recognition to reduce redundancy for extractive summarization task. Our work is partially inspired by the models of Pasunuru et al. (2017) with following differences: Pasunuru et al. (2017) model the entailment task as the seq2seq generation problem and enforce sharing of the same decoder between summarization and entailment. However, the entailment task is more reasonable to be considered as a multi-label classification problem rather than a generation problem. We thus design a multi-task learning framework in which the summarization generation task shares the same encoder with the entailment recognition task.

[2]We adopt the implement at https://github.com/pcyin/pytorch_nmt

[3]Multi-Genre Natural Language Inference (MultiNLI) is one of the largest corpora available for the task of natural language inference. It consists of sentences from ten different sources of text, which can be used for cross-genre domain adaptation.

5 Dataset

We conduct experiments on English Gigaword and DUC 2004 datasets.

Gigaword Corpus. We use the annotated Gigaword corpus provided by Rush et al. (2015). The dataset has about 3.8 million training pairs. Following Zhou et al. (2017), we use 8, 000 pairs as validation set and the test samples provided by Rush et al. (2015) and Zhou et al. (2017) as our test sets.

DUC 2004 Corpus. DUC-2004 corpus for tasks 1 & 2 (Over et al., 2007) consists of 500 documents. Each document in these datasets has four human annotated summaries. For experiments on this corpus, we directly use the model trained on the Gigaword to test on the DUC 2004 corpus.

6 Experiment

6.1 Experimental Settings

Word embedding size is set to 300 and LSTM hidden state size is set to 512. We use the full source and target vocabularies collected from the training data, which have 119, 505 and 68, 885 words, respectively. Adam (Kingma and Ba, 2014) optimizer is applied with the learning rate of 0.001, momentum parameters $\beta_1 = 0.9$ and $\beta_1 = 0.999$, and $\epsilon = 10^{-8}$. For RAML, $\tau = 0.85$. The mini-batch size is set to 64. We test the model performance (ROUGE-2 F1 score) on validation set for every 2,000 batches. We halve the learning rate if the ROUGE-2 F1 score drops for twelve consecutive tests on validation set. We also apply gradient clipping (Pascanu et al., 2012) with range $[-5, 5]$ during training. Our model with entailment-aware encode requires less than 300,000 training iterations to train with early stopping (Prechelt, 1998). To speed up the training for our RAML model, we continue the RAML training based on the pre-trained model with ML training with the current decayed learning rate. At test time, we use beam search with beam size 10 to generate the summary. We report ROUGE F1 score including ROUGE-1, ROUGE-2 and ROUGE-L for Gigaword corpus and ROUGE recall score for DUC 2004 corpus.

6.2 Comparative Methods

We compare a set of sentence summarization baselines.

ABS. Rush et al. (2015) first apply the seq2seq model to abstractive sentence summarization. They use an attentive CNN encoder and neural network language model decoder to summarize sentence.

ABS+. Rush et al. (2015) further tune ABS model on DUC 2003 dataset, then test on DUC 2004 test set.

CAs2s. Chopra et al. (2016) extend the ABS model with a convolutional encoder and RNN decoder, which performs better than the ABS model.

Feats2s. Nallapati et al. (2016) use a full RNN seq2seq model and add some lexical features to enhance the encoder, including POS, NER tags and TF-IDF values.

Luong-NMT. Luong et al. (2015) propose a neural machine translation model with two-layer LSTMs for the encoder-decoder.

Seq2seq. This is a standard seq2seq model with attention mechanism.

Seq2seq + MTL. This is our proposed model with entailment-aware encoder, which applies a multi-task learning (MTL) framework to seq2seq model.

Seq2seq + MTL (Share decoder). Pasunuru et al. (2017) propose a multi-task learning (MTL) framework in which the decoder is shared for summarization generation and entailment generation task.

Seq2seq + ERAML. This is our proposed model with entailment-aware decoder, which conducts an Entailment Reward Augmented Maximum Likelihood (ERAML) training framework.

Seq2seq + ROUGE-2 RAML. We apply ROUGE-2 RAML training for seq2seq model.

Seq2seq + RL. We implement Reinforcement Learning (RL) models (policy gradient) with reward metrics of **Entailment** and **ROUGE-2**.

Seq2seq + selective. Zhou et al. (2017) employ a selective encoding model to control the information flow from encoder to decoder. To verify the generalization of our entailment-based strategies, we adopt selective encoding mechanism to our seq2seq model and apply MTL and RAML to **Seq2seq + selective** model, which is denoted as the **Seq2seq + selective + MTL + RAML** model.

Model	ROUGE-1	ROUGE-2	ROUGE-L
ABS (Rush et al., 2015)	37.41	15.87	34.70
Seq2seq (Zhou et al., 2017)	43.76	22.28	41.14
Seq2seq + MTL	45.11	23.87	42.50
Seq2seq + MTL (Share decoder) (Pasunuru et al., 2017)	44.69	22.91	42.04
Seq2seq + ERAML	44.71	23.74	42.11
Seq2seq + ROUGE-2 RAML	43.75	23.63	41.31
Seq2seq + RL (Entailment)	44.39	23.31	41.86
Seq2seq + RL (ROUGE-2)	43.55	22.97	41.01
Seq2seq + MTL + ERAML	45.36	24.12	42.74
Seq2seq + selective	45.58	24.02	42.88
Seq2seq + selective + MTL + ERAML	**46.28**	**24.60**	**43.47**

Table 1: Experimental results (%) on the English Gigaword test set of Zhou et al. (2017). Our models perform significantly better than baselines by the 95% confidence interval measured by the official ROUGE script.

6.3 Experimental Results: Gigaword Corpus

In Table 1, we report the ROUGE F1 score of our model and the baseline methods on the English Gigaword test set provided by Zhou et al. (2017). Our entailment-aware models outperform all baseline models by a large margin. Our final model, **Seq2seq + selective + MTL + ERAML**, achieves the best results, which improves 2.52 (%) ROUGE-1, 2.32 ROUGE-2 and 2.33 ROUGE-L over seq2seq model. Our seq2seq model with entailment-aware encoder (**Seq2seq + MTL**) surpasses the state-of-the-art seq2seq model of 1.35 ROUGE-1, 1.59 ROUGE-2, 1.36 ROUGE-L, and entailment-aware decoder (**Seq2seq + ERAML**) gains improvement of 0.95 ROUGE-1, 1.46 ROUGE-2, 0.97 ROUGE-L. Compared to the another MTL model via sharing decoder for entailment generation task (**Seq2seq + MTL (Share decoder)**), our MTL model (**Seq2seq + MTL**) has obvious ROUGE score gains. The **Seq2seq + ROUGE-2 RAML** model also shows promising performance, especially for ROUGE-2 score. RAML has a clear advantage over RL. In principle, RL samples from the model distribution, which slows down training and several tricks are needed to get better estimates of the gradient (Ranzato et al., 2015). The comparison to the model of **Seq2seq + selective** shows that our entailment-aware strategies are also useful for seq2seq model with selective encoding framework, which demonstrates the good generalization of our method. The results on English Gigaword test set provided by Rush et al. (2015) are shown in Table 2. Our model performs better than the previous works.

6.4 Experimental Results: DUC 2004 Test Corpus

We evaluate our model with the ROUGE recall score. The reference summaries of the DUC 2004 test set are fixed to 75 bytes and we set the maximum length of the summary to 18 following Zhou et al. (2017). In Table 2, experimental results also show our **Seq2seq + selective + MTL + ERAML** model achieves significant improvements over baseline models, surpassing Feats2s (Nallapati et al., 2016) by 0.98% ROUGE-1, 0.78% ROUGE-2 and 0.65% ROUGE-L without fine-tuning on DUC data.

6.5 Manual Evaluation

Next, we conduct a manual evaluation to inspect the correctness of the generated summaries. We randomly select 500 samples in the test set and employ five postgraduates to classify the generated summaries as correct (i.e., not contain wrong information) or not. As shown in Table 3, 60.6% of the summaries generated by seq2seq model are correct, and it rises to 69.4% and 74.2% for our model with selective encoding and entailment-aware strategies, respectively, which indicates the effectiveness of our model to generate a correct summary.

	Test set of Rush et al. (2015)			DUC 2004 test set		
Model	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
ABS (Rush et al., 2015)	29.55	11.32	26.42	26.55	7.06	22.05
ABS+ (Rush et al., 2015)	29.76	11.88	26.96	28.18	8.49	23.81
Feats2s (Nallapati et al., 2016)	32.67	15.59	30.64	28.35	9.46	24.59
CAs2s (Chopra et al., 2016)	33.78	15.97	31.15	28.97	8.26	24.06
Luong-NMT (Luong et al., 2015)	33.10	14.45	30.71	28.55	8.79	24.43
Seq2seq (Zhou et al., 2017)	34.04	15.95	31.68	28.13	9.25	24.76
Seq2seq + MTL	34.69	16.68	32.32	28.61	9.67	24.86
Seq2seq + ERAML	34.34	16.59	32.26	28.37	9.61	24.81
Seq2seq + MTL + ERAML	34.88	16.86	32.51	28.89	9.87	24.94
Seq2seq + selective	35.01	16.71	32.88	29.01	9.89	25.01
Seq2seq + selective + MTL + ERAML	**35.33**	**17.27**	**33.19**	**29.33**	**10.24**	**25.24**

Table 2: Experimental results (%) on the English Gigaword test set of Rush et al. (2015) and DUC 2004 test set. Our models perform significantly better than baseline models by the 95% confidence interval measured by the official ROUGE (RG) script.

Model	Correctness(%)
Seq2seq	60.6
Seq2seq + selective	69.4
Seq2seq + selective + MTL + ERAML	74.2

Table 3: Manual evaluation for correctness.

6.6 Further Analysis

To further investigate the effectiveness of our model, we perform analysis on the entailment score improvement, the abstraction degree of our model and the impact for entailment recognition task.

6.6.1 Does our summarization model learn entailment knowledge?

The motivation of our work is to encourage summarization model to generate summaries that are entailed by the source sentences. To verify this goal, we investigate the entailment score for source-summary pairs for different models. For the test set of Zhou et al. (2017), the average entailment score for the reference is 0.72, while for the basic seq2seq model, the entailment score is only 0.46. When we adopt entailment-based strategies, the entailment score rises to 0.63 for seq2seq model. Note that the entailment score is 0.57 for seq2seq model with selective encoding, and we believe that the selective mechanism can filter out secondary information in the input, which will reduce the possibility to generate irrelevant information. Entailment-aware selective model achieves a high entailment reward of 0.71. In part at least, we can conclude that our model has successfully learned entailment knowledge.

6.6.2 Is it less abstractive for our model?

We have shown that our entailment-aware model can generate correct summaries more frequently (Section 6.5). Intuitively, it is more likely to be correct if summary segments are directly extracted from the source. Thus, readers may wonder whether our model is less abstractive. Figure 2 shows that the seq2seq model produces more novel words (i.e., words that do not appear in the article) than our model, indicating a lower degree of abstraction for our model. However, when we exclude all the words not in the reference (these words may lead to wrong information), our model generates more novel words, suggesting that our model provides a compromise solution for informativeness and correctness. Thus, our model can generate summary with fewer mistakes.

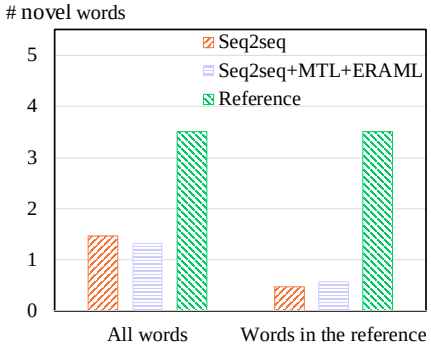


Figure 2: Average count of novel words (words that do not appear in the article). Seq2seq model generates more novel words, but less words are in the reference compared to our model.

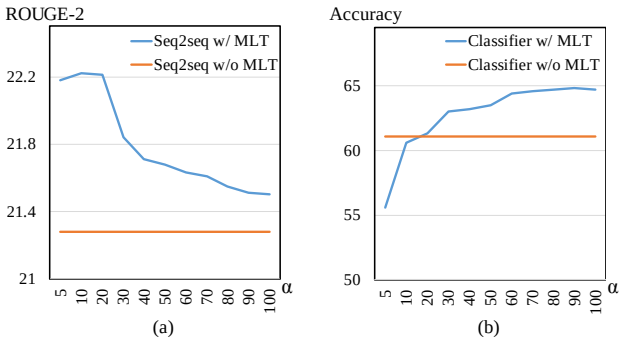


Figure 3: The performance of (a) summarization generation on Gigaword validation set and (b) entailment recognition on SNLI (Bowman et al., 2015) validation set with different task batch switches (α).

1	Source	brazilian stocks rose , led by consumer stocks , after the government said it would n't impose restraints on consumer credit
	Reference	brazil stocks rise after government rules out credit restraints
	Seq2seq	brazil stocks rise on consumer credit concerns
	Seq2seq + MTL + ERAML	brazil stocks rise after government says it wo n't impose credit restrictions
2	Source	authorities have denied neo-nazi groups permission to stage a demonstration next week in the austrian capital , where skinheads planned to gather on the ##th anniversary of nazi germany 's surrender ending world war ii in europe .
	Reference	vienna police deny neo-nazi groups permission to stage downtown
	Seq2seq	neo-nazi group to stage demonstration in vienna
	Seq2seq + MTL + ERAML	austrian authorities deny neo-nazi demonstration
3	Source	queens taxi driver osman chowdhury said he never thought of keeping the ## diamond rings he found inside a suitcase left in his trunk by a dallas woman who had given him a ##-cent tip , local media reported on thursday .
	Reference	u.s. taxi driver says he never thinks of keeping diamond rings left in trunk
	Seq2seq	queens taxi driver denies keeping ## diamond rings in trunk
	Seq2seq + MTL + ERAML	## diamond rings found in suitcase

Table 4: Cases Study.

6.6.3 Could the entailment recognition also be improved?

Multi-task learning (MTL) involves sharing parameters between related tasks, whereby each task can benefit from extra information of other tasks in the training process. In this section, we explore whether the entailment recognition can benefit from summarization generation task. Figure 3 shows that our summarization model with MTL outperforms basic seq2seq model. As α increases, the accuracy of entailment recognition improves and finally exceeds that of the model without MTL, which reveals the advantage of MTL framework.

7 Case Study

We illustrate the examples of outputs in Table 4. As shown in the table, seq2seq model generates summaries that are not relevant to the source sentence, while the output of our model obtains higher entailment scores than those of seq2seq model. For the first example, seq2seq model regards the reason for "brazil stocks rise" as "consumer credit concerns", while in fact, "consumer" is not worried because "government said it would n't impose restraints on consumer credit". By contrast, since our model incorporates entailment knowledge, the true reason is captured and the output of our model is related to the source sentence. A similar problem happens in example 2, and seq2seq model generates a summary that is contradictory to the source. The "demonstration" is "denied" by the "authorities", while seq2seq model confirms the "demonstration". In Example 3, neither seq2seq nor our model performs satisfactorily. Seq2seq model again misunderstands the meaning of the source and outputs summary containing wrong information. Though the summary generated by our model is entailed by the source, the summary fails to produce an integrated sentence and misses the key points of the source, such as the object of the event, "queens taxi driver". A mixed reward, i.e., combining entailment and ROUGE-2, may address this

issue. We leave it for our future work.

8 Conclusion

This paper investigates the correctness problem in abstractive summarization. We propose an entailment-aware encoder by jointly learning summarization generation and entailment recognition. We present an entailment-aware decoder by entailment reward augmented maximum likelihood training. By enriching the encoder and decoder with entailment information, our model makes the summary more likely be entailed by the source input. Experimental results on Gigaword and DUC 2004 datasets demonstrate that our model achieves significant improvements over strong baselines on both informativeness and correctness. Our code is available online[4].

Acknowledgements

The research work descried in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2017YFC0820700 and the Natural Science Foundation of China under Grant No. 61333018.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, pages 1137–1155.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. *arXiv:1711.04434*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *journal of artificial intelligence research*, 22:457–479.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 140–149.

Anand Gupta, Manpreet Kaur, Shachar Mirkin, Adarsh Singh, and Aseem Goyal. 2014. Text summarization through entailment-based minimum vertex cover. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 75–80.

[4]https://github.com/bubei/entail_sum

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2017a. Multi-modal summarization for asynchronous collection of text, image, audio and video. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1092–1102. Association for Computational Linguistics.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017b. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.

Haoran Li, Junnan Zhu, Tianshang Liu, Jiajun Zhang, and Chengqing Zong. 2018. Multi-modal sentence summarization with modality attention and image filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Shuming Ma, Xu Sun, Jingjing Xu, Houfeng Wang, Wenjie Li, and Qi Su. 2017. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 635–640.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and N. G. Raymond T. 2013. Abstractive meeting summarization with entailment and fusion. In *European Workshop on Natural Language Generation*, pages 136–146.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 130–136.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. In *Information Processing & Management*, pages 1506–1520.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. 2017. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 27–32.

Lutz Prechelt. 1998. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv:1511.06732*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1054–1059.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, pages 2903–2908.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv:1704.05426*.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. Efficient summarization with read-again and copy mechanism. *arXiv:1611.03382*.

Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2016. Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(10):1842–1853.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1104.